Light Up

There are three ways a solution can fail:

- If a light bulb shines on another light bulb
- If a blocked cell with a number has the wrong number of light bulbs around it
- If an open cell doesn't get lit

So, this problem is just some 2D array manipulation.

- For every cell:
  - If it's a light bulb ('?'), keep going up, down, left, right, marking open cells
    - If you see another light bulb, FAIL
  - If it's a number ('0', '1', '2', '3', '4') look 1 cell up, down, left, right, count the light bulbs
    - If the count doesn't match the number, FAIL

After that, go through the grid again.

- For every cell:
  - If it's an unmarked open cell, FAIL

Here's some Java code:

```
boolean ok = true;
for( int i=0; i<n; i++ ) for( int j=0; j<n; j++ )
{
  if( grid[i][j]=='?' )
  {
    int d;
    for( d=i-1; d>=0 && (grid[d][j]=='.' || grid[d][j]=='#'); --d ) grid[d][j] = '#';
    if( d>=0 && grid[d][j]=='?' ) ok = false;
    for( d=i+1; d<n && (grid[d][j]=='.' || grid[d][j]=='#'); ++d ) grid[d][j] = '#';
    if( d<n && grid[d][j]=='?' ) ok = false;
    for( d=j-1; d>=0 && (grid[i][d]=='.' || grid[i][d]=='#'); --d ) grid[i][d] = '#';
    if( d>=0 && grid[i][d]=='?' ) ok = false;
    for( d=j+1; d<n && (grid[i][d]=='.' || grid[i][d]=='#'); ++d ) grid[i][d] = '#';
    if( d<n && grid[i][d]=='?' ) ok = false;
  }
  else if( Character.isDigit( grid[i][j] ) )
  {
    int constraint = grid[i][j] - '0';
    int count = 0;
    if( i-1>=0 && grid[i-1][j]=='?' ) ++count;
    if( i+1<n && grid[i+1][j]=='?' ) ++count;
    if( j-1>=0 && grid[i][j-1]=='?' ) ++count;
    if( j+1<n && grid[i][j+1]=='?' ) ++count;
    if( count!=constraint ) ok = false;
  }
}
for( int i=0; i<n; i++ ) for( int j=0; j<n; j++ ) if( grid[i][j]=='.' ) ok = false;
ps.println( ok ? 1 : 0 );
```